Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 2543824, 16 pages
https://doi.org/10.1155/2020/2543824

Research Article

C^2-Continuous Orientation Planning for Robot End-Effector with B-Spline Curve Based on Logarithmic Quaternion

Yasong Pu,[1,2] **Yaoyao Shi**,[1] **Xiaojun Lin**,[1] **Yuan Hu**,[1] **and Zhishan Li**[1]

[1]*School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, China*
[2]*College of Engineering, Honghe University, Mengzi 661100, China*

Correspondence should be addressed to Yasong Pu; puyasong@mail.nwpu.edu.cn

Received 4 June 2020; Accepted 26 June 2020; Published 22 July 2020

Academic Editor: Gaetano Giunta

Copyright © 2020 Yasong Pu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Smooth orientation planning is beneficial for the working performance and service life of industrial robots, keeping robots from violent impacts and shocks caused by discontinuous orientation planning. Nevertheless, the popular used quaternion interpolations can hardly guarantee C^2 continuity for multiorientation interpolation. Aiming at the problem, an efficient quaternion interpolation methodology based on logarithmic quaternion was proposed. Quaternions of more than two key orientations were expressed in the exponential forms of quaternion. These four-dimensional quaternions in space S^3, when logarithms were taken for them, could be converted to three-dimensional points in space R^3 so that B-spline interpolation could be applied freely to interpolate. The core formulas that B-spline interpolated points were mapped to quaternion were founded since B-spline interpolated point vectors were decomposed to the product of unitized forms and exponents were taken for them. The proposed methodology made B-spline curve applicable to quaternion interpolation through dimension reduction and the high-order continuity of the B-spline curve remained when B-spline interpolated points were mapped to quaternions. The function for reversely finding control points of B-spline curve with zero curvature at endpoints was derived, which helped interpolation curve become smoother and sleeker. The validity and rationality of the principle were verified by the study case. For comparison, the study case was also analyzed by the popular quaternion interpolations, *Spherical Linear Interpolation* (SLERP) and *Spherical and Quadrangle* (SQUAD). The comparison results demonstrated the proposed methodology had higher smoothness than SLERP and SQUAD and thus would provide better protection for robot end-effector from violent impacts led by unreasonable multiorientation interpolation. It should be noted that the proposed methodology can be extended to multiorientation quaternion interpolation with higher continuity than the second order.

1. Introduction

Trajectory planning of industrial robot end-effector in workspace is a fundamental and important task. Smooth and nonimpact trajectory planning is necessary for the working performance and the service life of industrial robots, otherwise unsmooth trajectory planning will bring added violent impacts and shocks to the robot degrading robot's work. The trajectory planning of industrial robots is divided into position planning and orientation planning, while the valid and reasonable orientation planning is one of the foundational cores to ensure perfect working completion for the robot mechanical arm. Orientation planning is related to rotation arrangement and is a broad and complicated project [1]. Lots of relevant researchers put great efforts into orientation planning and lots of methodologies were suggested [2–6].

The main expressions of orientation of a rigid body are Euler angle, rotation matrix, and quaternion [7]. Quaternion used to describe orientation has the following advantages: (1) avoid the phenomenon of locking universal joints when rotating; (2) higher calculation efficiency; (3) easy to provide smooth interpolation. Because of the remarkable characteristics, quaternion has been widely used in rotating applications. However, quaternion also has its disadvantages in describing orientation. Quaternion belongs to rotation space and is a four-dimensional vector. As a result, the algorithms of the three-dimensional vector in Cartesian space are not

suitable for quaternion directly [8], and interpolating theories in the three-dimensional vector space cannot be directly applied to quaternion orientation interpolation. That is, high-order interpolation methods in Cartesian space cannot be directly used for multiorientation interpolation of quaternion. Though many scholars have been proposing kinds of quaternion orientation interpolations, the existing research for quaternion interpolation still needs to be developed further especially for smooth higher-order continuous interpolation between multiple orientations.

The most widely used orientation interpolation of quaternion is *Spherical Linear Interpolation* (SLERP) raised by Shoemake. The reasoning process of SLERP is concise and uncomplicated, and the application of SLERP is simple and convenient. The formula of SLERP is as the following formula [9]:

$$\mathrm{SLERP}(Q_1, Q_2, t) = \frac{\sin[(1-t)\theta]}{\sin(\theta)} Q_1 + \frac{\sin(t\theta)}{\sin(\theta)} Q_2 t \in [0,1], \tag{1}$$

where Q_1 is the original orientation, Q_2 is the end orientation, and θ is the angle between the 4-dimesion vectors of Q_1 and Q_2. The angle value of θ can be gotten by the dot product $\cos(\theta) = Q_1.Q_2$. A series of smooth quaternion orientations between Q_1 and Q_2 can be obtained when the variable t changes uniformly between 0 and 1.

SLERP is very suitable for quaternion interpolation between only two orientations. However, when interpolations between more than two orientations proceed, SLERP can only guarantee C^0 continuity, and there would be sharp points at connecting points which connect several segments of interpolations. That means trajectory of orientation interpolation of quaternion is unsmooth at connecting points. As a result, the angular velocity and the angular acceleration would have discontinuous breakpoints which will bring violent impacts and shocks to industrial robots. Therefore, SLERP is not suitable for multiorientation interpolation for industrial robots.

In order to acquire smooth quaternion interpolation between more than two orientations, Shoemake presented *Spherical and Quadrangle* (SQUAD) developed on the base of SLERP. Although the derivation and application of SQUAD seem more complicated, it is still a matured algorithm and is often used for smooth quaternion interpolation between more than two orientations.

The formula of SQUAD is as the following formula [10]:

$$\begin{cases} \mathrm{SQUAD}(Q_i, Q_{i+1}, S_i, S_{i+1}, h) = \mathrm{SLERP}[\mathrm{SLERP}(Q_i, Q_{i+1}, h), \mathrm{SLERP}(S_i, S_{i+1}, h), 2h(1-h)], \quad i = 0, 1, \ldots, n-1 h \in [0,1], \\ S_i = Q_i * \exp\left(-\dfrac{\ln(Q_i^{-1} * Q_{i-1}) + \ln(Q_i^{-1} * Q_{i+1})}{4}\right), \quad i = 0, 1, \ldots, n-1. \end{cases} \tag{2}$$

Formula (2) has two functions. The first function is the interpolation function, and the second is called auxiliary function. $Q_0, Q_1, \ldots, Q_n$ are series of key orientations given in the form of quaternion, h is the only variable, Q_i^{-1} is the inverse quaternion of Q_i, and S_i and S_{i+1} are called auxiliary quaternions which can be gotten by the second function in formula 2. If we want to obtain the values of S_0 and S_n in auxiliary function, the values of Q_{-1} and Q_{n+1} are needed. However, we only have the values of $Q_0, Q_1, \ldots, Q_n$. So, we set the value of S_0 equal to that of Q_0, and the value of S_n equal to that of Q_n. This setting has little influence on the resulting interpolation curve. After the SLERP formula is used three times, SQUAD is finished.

Dam et al. proved that the formula SQUAD has C^1 continuity of quaternion interpolation [10] and showed by the figure that both the angular and the angular velocity in the interpolation examples by SQUAD were continuous. However, they did not mention whether SQUAD has the C^2-continuity characteristic. The subsequent case example in Section 6 of this paper shows that there are broken points at connecting nodes in the angular acceleration figure obtained by SQUAD. That means SQUAD is not C^2 continuous at connecting points for multiorientation quaternion interpolation. Discontinuous angular acceleration will also bring violent impacts on industrial robots and may degrade its working performance and service life as well.

In order to improve high-order smoothness of multiorientation interpolation of quaternion, some scholars introduce quaternion spline curves into quaternion interpolation. Myoung-jun Kim et al. presented a kind of quaternion spline curve with higher-order derivability [11]. They proposed the concept of cumulative basis function of quaternion, and the cumulative basis function of quaternion is taken as the exponent to perform quaternion interpolation. The operation of cumulative basis function becomes more complicated since its calculation is involved quaternion. Moreover, when quaternion B-spline curve is used for orientation interpolation, the control vertexes need be found reversely by the given key orientation quaternions. However, since the quaternion curve is generated in non-Euclidean space, the control vertexes obtained by reverse solution needs to solve nonlinear equations, so the numerical solution of the control vertexes of the quaternion B-spline curve can only be obtained approximately.

Yan Xing et al. presented a class of the generalized quaternion curve of B-spline [12]. They constructed series of nonlinear core functions based on the research findings of Juhász and Róth. More paths of rotation can be attained by

means of choosing different core functions. They concluded the classical unit quaternion curves of B-spline in these rotating paths and considered the classical unit quaternion curves of B-spline as special cases. And the generalized B-spline quaternion curves can achieve higher order of continuity. Applications and calculations of the research findings seem a little more complex.

Although there are also other several ways to construct the high-order quaternion curve [13–18], it is still difficult to tell which one method is better than another, and there is still no perfect solution so far. Relevant researchers have been striving to study and explore in this direction. So, the purpose of this paper is to find a compact and effective method of smooth multiorientation interpolation of quaternion.

2. Quaternion Orientation Interpolation Principle with Spline Curve Based on Logarithmic Quaternion

2.1. Preliminaries of Quaternion. When a rigid body rotates purely, both the orientation of the rigid body changes and the position of the particle on the rigid body change. A rotation means a new orientation of the rigid body. Conversely, knowing the orientation of a rigid body means knowing what rotation the rigid body has made. That is to say, a rotation equals an orientation.

The main expressions of orientation (i.e., rotation) are as follows: rotation matrix, Euler angle, and quaternion. The rotation matrix is composed of nine elements, which results in large data and inefficiency in calculation compared with other two methods. What is more, the rotation matrix is not convenient in smooth interpolation. The big trouble with the Euler angle is the phenomenon of *Gimbal Lock*. That is, there will be singular points under some conditions when the second Euler angle is the multiple of 90° [19]. When *Gimbal Lock* happens, movement will not follow the expected path. By comparison, quaternion has less computational complexity, has not the phenomenon of *Gimbal Lock*, and is convenient for smooth interpolation so that it is widely used in the representation of rotation (i.e., orientation). Although quaternion has many advantages, it also has its weaknesses. Quaternion is a four-dimensional vector and cannot be visually represented in Cartesian space. Moreover, the algorithms of three-dimensional vector operation and interpolation laws cannot be directly applied for quaternion. So, these problems bring some difficulties to quaternion orientation interpolation, especially for high-order continuous orientation interpolation. Thus, lots of research are involved in it.

Quaternion can be generally defined as $Q = q_0 + q_1\mathrm{i} + q_2\mathrm{j} + q_3\mathrm{k}$, where, q_0, q_1, q_2, and q_3 are all real numbers and i, j, and k are the imaginary axes of the complex number. Quaternion can also be represented as the vector form $Q = [q_0, (q_1, q_2, q_3)] = [w, v]$, where w is called the real part of quaternion and v is called the imaginary part of quaternion. The modulus of quaternion can be expressed as $\|Q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$.

When the real part of a quaternion Q is zero, such quaternion is called pure quaternion. Pure quaternion can be expressed as $Q = [0, (q_1, q_2, q_3)] = [0, v]$; when the imaginary part of the quaternion Q becomes the opposite values, such quaternion is named conjugate quaternion and is expressed in Q^*, and $Q^* = [q_0, -(q_1, q_2, q_3)] = [w, -v]$. The modulus of quaternion can also be expressed by $\|Q\| = \sqrt{QQ^*}$; when the multiplication of two quaternions Q_1 and Q_2 is equal to [1,0,0,0], Q_2 is defined as the inverse quaternion of Q_1. Inverse quaternion of Q is expressed in Q^{-1}, and $Q^{-1} = Q^*/\|Q\|^2$. When the modulus of a quaternion is equal to 1 ($\|Q\| = 1$), such quaternion is called unit quaternion. Nonunit quaternion can be converted to unit quaternion by $Q/\|Q\|$. For a unit quaternion, its conjugate quaternion Q^* is equal to its inverse quaternion Q^{-1}; that is, $Q^* = Q^{-1}$. Because the modulus of any unit quaternion is equal to 1 ($\|Q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1$), unit quaternion belongs to S^3 field. Quaternion used to describe rotation is usually unit quaternion.

Compared with the mathematical operations of real number, complex number, and vector, the mathematical operations of quaternion is more complicated. And the commutative law is not appropriate for quaternion operation. If there are two arbitrary quaternions H ($H = [h_0, u]$) and Q ($Q = [q_0, v]$), the multiplication operation is as the following equation:

$$\mathrm{HQ} = [h_0, u] * [q_0, v] = h_0 q_0 - u \cdot v + h_0 v + q_0 u + u \times v. \tag{3}$$

Notice that HQ is usually not equal to QH.

In general, quaternion used for rotation is mostly unit quaternion. Furthermore, unit quaternion can be written as formula (3), which is more popular than original definition expression:

$$Q = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right) * (a, b, c)\right] = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right) * n\right], \tag{4}$$

where θ is the rotation angle and n=(a,b,c) is the vector of the rotation axis. In addition, n is the unit vector ($\|n\| = 1$). Function (4) shows the rotation characteristic of quaternion; that is, a quaternion means a rotation or an orientation.

2.2. Deduction of Quaternion Interpolation Principle Based on Logarithmic Quaternion. As we know, the common complex number F can be defined as $F = A + B\mathrm{i}$. The angle of a complex number can be expressed by $\varphi = \arctan(B/A)$. The complex, whose modulus equals 1 ($\|F\| = 1$), is called the unit complex. The unit complex can be written in the exponential form as $F = e^{\phi i}$. Quaternion is also a kind of the complex number, which is more complicated than the plane complex number. In the same way, unit quaternion can be expressed in exponential form similar to the common complex:

$$Q = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right) * n\right] = e^{\theta n} = e^{\theta * (a,b,c)}, \tag{5}$$

where the meanings of θ and n are the same as that in function (4).

When we take logarithms on both sides of equation (5), we get

$$\ln(Q) = \theta n = \theta * (a, b, c) = (x, y, z) = P. \tag{6}$$

As can be seen from equation (6), if we take a logarithm for a unit quaternion, the four-dimensional quaternion in S^3 space can be converted to three-dimensional points in R^3 space. At this time, high-order interpolation theories in Cartesian space become available, such as B-spline interpolation especially. Suppose there are start orientation and end orientation of robot end-effector, which are described as quaternion orientations Q_s and Q_e, respectively. If the logarithm is taken for Q_s and Q_e according to equation (5), we can accordingly get three-dimensional start point P_s and end point P_e in Cartesian space. Then, we can use various interpolation methods in R^3 space to interpolate and get series of interpolated points marked as P_i=(x_i, y_i, z_i) and $i = 0$, 1, 2, . . ., n. According to relevant interpolation equation, the interpolated points can be written as follows:

$$P(t) = [x(t), y(t), z(t)], \quad t \in [0, 1]. \tag{7}$$

The interpolated points are three-dimensional vectors in R^3 space and need to be converted to unit quaternions in S^3 space. To ensure the rotation axes of the quaternions are unit vectors, the rotation axis vectors should be united. Thus, the interpolation equation should be decomposed to the product of unitized forms as follows:

$$P(t) = \sqrt{x^2(t) + y^2(t) + z^2(t)} \cdot \frac{[x(t), y(t), z(t)]}{\sqrt{x^2(t) + y^2(t) + z^2(t)}}, \quad t \in [0, 1]. \tag{8}$$

Substitute $P(t)$ in equation (8) into equation (6), and take exponents on both sides of the equation. We can get

$$Q(t) = e^{p(t)} = e^{\left\{ \sqrt{x^2(t)+y^2(t)+z^2(t)} \cdot \frac{[x(t), y(t), z(t)]}{\sqrt{x^2(t)+y^2(t)+z^2(t)}} \right\}}. \tag{9}$$

By comparing equation (9) with equation (5), the rotation angle and rotation axis vector of interpolated unit quaternion can be obtained as follows:

$$\begin{cases} \theta(t) = \sqrt{x^2(t) + y^2(t) + z^2(t)}, \\ a(t) = \dfrac{x(t)}{\sqrt{x^2(t) + y^2(t) + z^2(t)}}, \\ b(t) = \dfrac{y(t)}{\sqrt{x^2(t) + y^2(t) + z^2(t)}}, \\ c(t) = \dfrac{z(t)}{\sqrt{x^2(t) + y^2(t) + z^2(t)}}, \end{cases} \quad t \in [0, 1], \tag{10}$$

where $\theta(t)$ and $[a(t), b(t), c(t)]$, respectively, are the rotation angle and the rotation axis vector of interpolated unit quaternion in S^3. $[x(t), y(t), z(t)]$ is the vector of the interpolated point in R^3.

Finally, the real part and complex part of interpolated unit quaternion can be deduced as follows:

$$Q(t) = \begin{cases} q_0(t) = \cos\left(\dfrac{\sqrt{x^2(t) + y^2(t) + z^2(t)}}{2} \right), \\ q_1(t) = \sin\left(\dfrac{\sqrt{x^2(t) + y^2(t) + z^2(t)}}{2} \right) \cdot \dfrac{x(t)}{\sqrt{x^2(t) + y^2(t) + z^2(t)}}, \\ q_2(t) = \sin\left(\dfrac{\sqrt{x^2(t) + y^2(t) + z^2(t)}}{2} \right) \cdot \dfrac{y(t)}{\sqrt{x^2(t) + y^2(t) + z^2(t)}}, \\ q_3(t) = \sin\left(\dfrac{\sqrt{x^2(t) + y^2(t) + z^2(t)}}{2} \right) \cdot \dfrac{z(t)}{\sqrt{x^2(t) + y^2(t) + z^2(t)}}, \end{cases} \quad t \in [0, 1], \tag{11}$$

where $[q_0(t), q_1(t), q_2(t), q_3(t)]$ is interpolated unit quaternion in S^3. $[x(t), y(t), z(t)]$ is a vector of the interpolated point in R^3. As shown in formula (11), the interpolated points in R^3 space can be mapped to the corresponding interpolated unit quaternions in S^3 space.

Make a summery here. The four-dimensional unit quaternions in S^3 space, when logarithms were taken for them, could be converted to three-dimensional points in R^3 space. Then, three-dimensional spline curve interpolation, such high-order smooth curve interpolation as the B-spline

curve, can be applied freely. When interpolation is completed in R^3 space, the three-dimensional interpolated points in R^3 space can be mapped to the interpolated unit quaternions in S^3 space according to formula (11). Thus, quaternion interpolations are realized.

It should be pointed out that the rotation angle θ in formula (10) is positive. When the rotation angle θ of the key orientation given in advance is negative, it can be dealt as follows. The rotation angle θ is taken as absolute value, and the rotation axis vector is taken as the opposite number before interpolation. That is, $\theta^*(a,b,c)$ are treated as $|\theta| * (-a, -b, -c)$.

From equations (4) and (5), we can obtain the following equation:

$$Q = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\left(\frac{\theta}{2}\right)\right) * n\right] = e^{\theta n} = e^{(x,y,z)}. \tag{12}$$

Equation (12) can be written as follows:

$$Q(t) = e^{[x(t),y(t),z(t)]}. \tag{13}$$

In equation (13), $[x(t), y(t), z(t)]$ are interpolated points of the spline curve in Cartesian space and can be expressed as $r(t)$. When we take exponent for $r(t)$ and get a new equation, the continuity of the new function is consistent with the one of $r(t)$; that is, the continuity of the new function will not change. For example, if we choose the cubic B-spline curve to be the interpolation method, the interpolated points are C^2 continuous at knot points. When exponent is taken for the function of the cubic B-spline curve and new function is obtained, continuity of new function will not change and remain second-order C^2 continuity. Taking exponent is the process of mapping the spline curve in Cartesian space to quaternion space, so the corresponding quaternions also keep the corresponding continuity, and the order of continuity remains unchanged.

3. B-Spline Curve Interpolation with Zero Curvature at Endpoints

3.1. Preliminaries of B-Spline Curve. The B-spline curve plays an important role in the freeform curve and surface modeling design and is a significant branch for smooth interpolation. The B-spline curve is developed on the basis of the Bezier curve. It retains the advantages of the Bezier curve and overcomes the defect that the Bezier curve has not the ability of local modification. Furthermore, it has advantages in dealing with complicated shapes. Because of its smooth interpolation and convenience, the B-spline curve has been one of the most popular methods of mathematical description of freeform curve and surface.

The mathematical definition of the B-spline interpolation curve is as follows [20]:

$$r(t) = \sum_{i=0}^{n} N_{i,k}(u) d_i, \tag{14}$$

where d_i are the control points and $N_{i,k}(u)$ are the basis functions. The label k is the power of basis function. The label i is the sequence label. Basis function of *de Boor Cox* is commonly used and is defined as follows:

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1 & u_i \le u \le u_{i+1} \\ 0 & \text{other cases,} \end{cases} \\ N_{i,k}(u) = \dfrac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \dfrac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u), \\ \text{Regulation : } \dfrac{0}{0} = 0, \end{cases} \tag{15}$$

where u_i are the nodes of the basis function.

The Cubic uniform B-spline curve is most widely used in engineering among kinds of B-spline curves because its expression is not very complicated and it has the properties of high smoothness and C^2 continuity, which can meet the needs of most engineering. At the same time, the cubic uniform B-spline also has the advantages of low power, simple calculation, fast speed, and the ability of constructing various shapes. As a result, numerous engineers and designers are favored of the cubic uniform B-spline curve when modeling freeform curve and surface.

The basis function of the cubic uniform B-spline curve can be described further in its effective regions of basis function nodes as follows [21]:

$$N_{i,3}(u) = \begin{cases} N_{0,3}(u) = \dfrac{1}{6}\left(-u^3 + 3u^2 - 3u + 1\right), \\ N_{1,3}(u) = \dfrac{1}{6}\left(3u^3 - 6u^2 + 4\right), \\ N_{2,3}(u) = \dfrac{1}{6}\left(-3u^3 + 3u^2 + 3u + 1\right), \\ N_{3,3}(u) = \dfrac{1}{6}u^3, \end{cases} \quad 0 \le u \le 1. \tag{16}$$

By synthesizing formula (14) and (16), practical formula of the cubic uniform B-spline curve can be obtained as the following formula [21]. Unless the other is specified, the following B-spline curves all refer to cubic uniform B-spline curves:

$$\begin{aligned} \mathbf{r}_i(u) &= \sum_{j=0}^{3} N_{j,3}(u) d_{i+j} \\ &= \frac{1}{6}\begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} d_i \\ d_{i+1} \\ d_{i+2} \\ d_{i+3} \end{bmatrix}, \quad u \in [0, 1], \end{aligned} \tag{17}$$

where d_{i+j} are the control points, $N_{j,3}(u)$ are the basis functions, and u is the variable.

Formula (17) is really practical for usage of the cubic uniform B-spline curve. The formula is easy to understand and convenient to use. As can be seen from formula (17), every four control points hold a B-spline curve. For example, if there are five control points, the two B-spline curves can be obtained, and so on. B-spline curves are C^2 continuous at knots between every two curves [22]. It should be pointed out that no matter how many control points there are, the basis functions are the same four functions as far as the cubic uniform B-spline curve is concerned.

As described above, the B-spline curve can be obtained if control points are given. However, in many engineering problems, we are often not given the control points but the knots which the spline curve would pass through. Knots are the key points belonging to the B-spline curve while control points are not a part of the curve so that we need to find control points for the B-spline curve. The process of finding control points by the use of knots is called inverse solution.

For the cubic uniform B-spline curve, the formula of inverse solution of control points is expressed as follows [21]:

$$\frac{1}{6}\left(d_{i\text{-}1}+4d_i+d_{i+1}\right)=P_i,\quad i=0,1,\ldots,\mathrm{n}, \tag{18}$$

where $d_{i\text{-}1}$, d_i, and d_{i+1} are the control points and P_i are the key points called as knots through which the B-spline curve must pass.

In function (18), the number of unknown control points is more than the number of equations, so there are numerous solutions for the equations. In order to obtain unique solutions of this function, we need to add two constraint equations. There are several common methods to add constraint equations. We need to search a better method suitable for the characteristic of quaternion interpolation.

3.2. The Unique Solution of Control Points with Zero Curvature at Endpoints. For the purpose that the interpolation curves are prevented from changing violently, we could consider that the curvatures at the beginning knot and the terminal knot of the interpolation curve are all set zero. We need add two constrain equations when the curvatures at the beginning and the terminal knots of the B-spline curve are all set to zero and can obtain the unique set of solutions of control points.

When the first and second control points are coincident, the curvature at the beginning knot of the B-spline curve will be zero, and the equation as below should hold [21]:

$$\overrightarrow{d_iP_{\mathrm{i}}}=\frac{1}{6}\times\overrightarrow{d_id_{i+1}}, \tag{19}$$

where d_{i-1}, d_i, and d_{i+1} are the first, the second, and the third control points, respectively. Control points d_{i-1} and d_i are coincident. P_i is the beginning knot of the B-spline curve. These relationships are clearly shown in Figure 1.

From equation (19), we can deduce one of the constraint equations as follows:

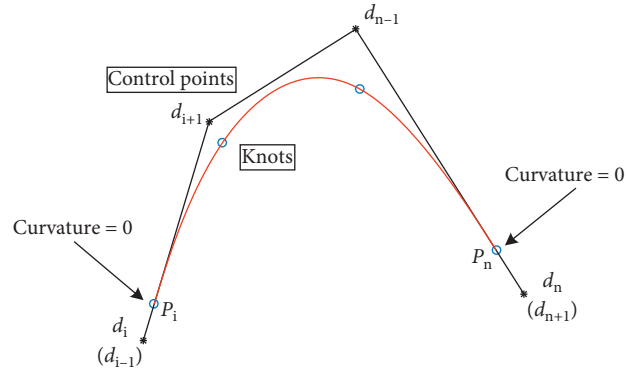


Figure 1: Cubic B-spline curve with free endpoint conditions.

$$\frac{5}{6}d_i+\frac{1}{6}d_{i+1}=P_i. \tag{20}$$

Similarly, when the last and the penultimate control points are coincident, the curvature at the terminal knot of the B-spline curve will be zero, and the equation as below should hold:

$$\overrightarrow{d_nP_n}=\frac{1}{6}\times\overrightarrow{d_nd_{n-1}}, \tag{21}$$

where $d_{n\text{-}1}$, d_n, and d_{n+1} are the last control point, the penultimate control point, and the third control from the bottom, respectively. Control points d_{n+1} and d_n are coincident. P_n is the terminal knot of the B-spline curve.

From equation (21), we can also deduce the other constrain equation as follows;

$$\frac{1}{6}d_{n-1}+\frac{5}{6}d_n=P_n. \tag{22}$$

By synthesizing equations (18), (20), and (22), we can finally obtain the following formula, which can be used to solve control points inversely by key knots of the B-spline curve under the condition that the curvatures at the beginning and the terminal knots of the B-spline curve are zero:

$$\frac{1}{6}\begin{bmatrix}5&1&0&0&0&\cdots&0\\1&4&1&0&0&\cdots&0\\0&1&4&1&0&\cdots&0\\0&0&1&4&1&0&\cdots\\&&&\vdots&&&\\0&0&\cdots&0&1&4&1\\0&0&0&\cdots&0&1&5\end{bmatrix}\begin{bmatrix}d_1\\d_2\\d_3\\d_4\\\vdots\\d_{n-1}\\d_n\end{bmatrix}=\begin{bmatrix}P_1\\P_2\\P_3\\P_4\\\vdots\\P_{n-1}\\P_n\end{bmatrix}. \tag{23}$$

In formula (23), d_1, d_2,. . ., d_{n} are a series of control points. The control point d_1 has a coincidence point d_0, which means the first control point has two identical control points d_1 and d_0. Similarly, the control point d_{n} has a coincidence point $d_{\mathrm{n+1}}$, and the last control point has two identical control points d_{n}

and d_{n+1}. The key knots P_1, P_2,. . .,P_n are given beforehand and must be passed through by the B-spline interpolation curve. The control points are the unknown and need to be solved. We can see from function (23) that the quantity of the unknown control points is equal to the equation quantity of the matrix so that function (23) has a unique set of solutions of control points. Then, we can interpolate between the key knots with cubic uniform B-spline curve after the values of control points are found.

4. Steps of Orientation Interpolation with B-Spline Curve Based on Logarithmic Quaternion

We sum up the steps of orientation interpolation with the B-spline curve based on logarithmic quaternion as follows:

Step 1: depending on the tasks of industrial robots, several key orientations of end-effector are obtained. The key orientations are expressed by unit quaternions Q_i, and rotation angles θ_i and rotation axes n_i can be gotten as well.

Step 2: according to functions (5) and (6), the four-dimensional unit quaternions Q_i are expressed in the exponential forms of the complex number and are transformed to three-dimensional vectors P_i in R^3 space when logarithms are taken for these quaternions. Then, interpolation theory of the B-spline curve becomes available.

Step 3: the obtained points P_i are considered as the key knots of B-spline interpolation curves, and we can find the control points d_i of the B-spline curve with zero curvature at end knot points by resolving function (23).

Step 4: according to the obtained control points d_i, we can use formula (17) to apply the B-spline curve to interpolate between every two key knots P_i, and get series of interpolated points $P(t)$ in R^3.

Step 5: all the interpolated points $P(t)$ of the B-spline curve in R^3 space are mapped to interpolated quaternions in S^3 space via formula (11). Quaternion interpolation ends.

The interpolation process can be demonstrated clearly by the following flowchart in Figure 2.

5. Case Study

In order to verify the correctness and rationality of the proposed methodology, the following study case is employed. The usage of the proposed methodology is further described in detail during the study case analysis.

Case 1: In order to accomplish a certain task for an industrial robot, the end-effector has to experience four necessary orientations. The first orientation is to rotate 60° around the Z axis in the fixed frame. The second orientation is to rotate 45° around the Y axis in the fixed frame. The third orientation is to rotate 90° around the X axis in the fixed frame. The last orientation is to rotate 90° around the axis whose vector is n=(2,3,5) in the fixed frame. Using the interpolation methodology proposed in this paper, complete quaternion orientation interpolation with guarantee of C^2 continuity.

Case analysis: there are four key orientations given in the case as seen in Figure 3. The multiorientation planning has three interpolation segments: the first segment between the 1st key orientation and the 2nd key orientation, the second segment between the 2nd key orientation and the 3rd key orientation, and the third segment between the 3rd key orientation and the 4th key orientation. The orientation interpolation process is as below. It should be pointed out that the B-spline curves used in the case are the cubic uniform B-spline curve.

(1) According to the four necessary orientations of industrial robots, we can get four key quaternions Q_i and then get the corresponding rotation angles θ_i and axes n_i, as seen in Table 1.

(2) According to formula (6), we can calculate the key knots P_i of the cubic uniform B-spline curve in R^3 Cartesian space. Depending on these key knots P_i obtained, we can find the resolutions of control points d_i by the use of formula (23), which ensures that the curvatures at the beginning and the terminal points of the B-spline curve are zero to avoid the curve changing violently. The values of key knots P_i and control points d_i are filled in Table 2.

(3) Using the control points d_i obtained, interpolations can be carried out between ever two key knots with cubic B-spline curves. According to B-spline interpolation formula (17), the interpolation functions of cubic B-spline curves can be obtained as below.

The first B-spine curve function:

$$r_1(u) = \begin{cases} x_1(u) = 0.791u^3 - 0.396u^2 - 0.396u, \\ y_1(u) = -0.914u^3 + 0.850u^2 + 0.850u, \\ z_1(u) = 0.510u^3 - 0.779u^2 - 0.779u + 1.04,7, \end{cases} \quad u \in [0,1]. \tag{24}$$

The second B-spine curve function:

$$r_2(u) = \begin{cases} x_2(u) = -1.595u^3 + 1.978u^2 + 1.187u, \\ y_2(u) = 1.299u^3 - 1.892u^2 - 0.193u + 0.785, \\ z_2(u) = 0.053u^3 + 0.752u^2 - 0.805u, \end{cases} \quad u \in [0,1]. \tag{25}$$

The third B-spine curve function:

$$r_3(u) = \begin{cases} x_3(u) = 1.343u^3 - 2.805u^2 + 0.360u + 1.571, \\ y_3(u) = -0.989u^3 + 2.005u^2 - 0.079u, \\ z_3(u) = -0.599u^3 + 0.911u^2 + 0.858u, \end{cases} \quad u \in [0,1]. \tag{26}$$

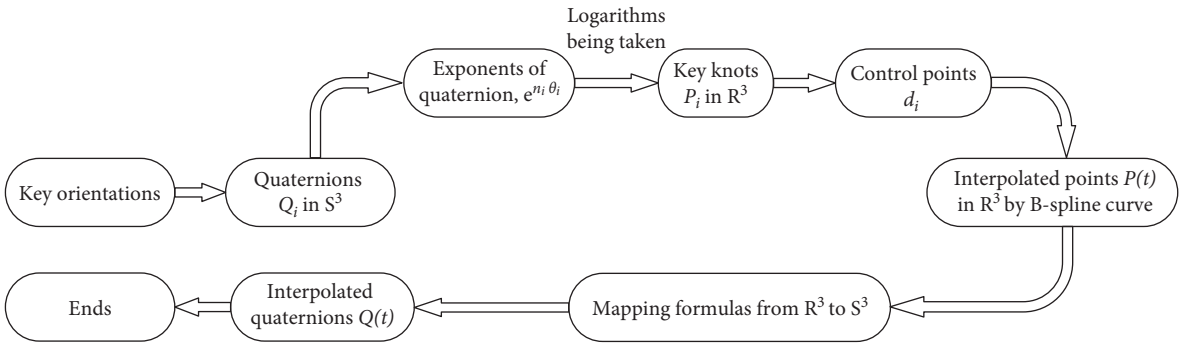

Logarithms being taken
Exponents of quaternion, $e^{n_i \theta_i}$
Key knots P_i in R^3
Control points d_i
Key orientations
Quaternions Q_i in S^3
Interpolated points $P(t)$ in R^3 by B-spline curve
Ends
Interpolated quaternions $Q(t)$
Mapping formulas from R^3 to S^3

Figure 2: Flowchart of quaternion interpolation with B-spline based on logarithmic quaternion.

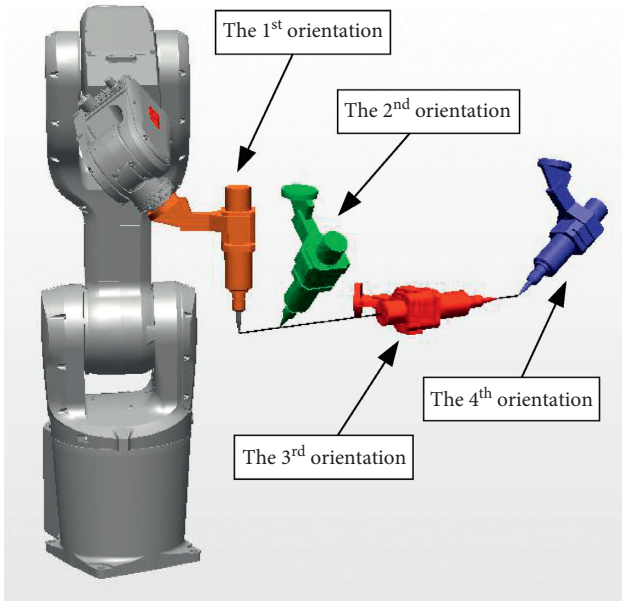


Figure 3: The four key orientations in working.

Table 1: Initial information of the four key orientations.

No.	Quaternion Q_i	Rotation angle θ_i	Rotation axis n_i
1	[0.866, 0, 0, 0.500]	$\pi/3$	(0, 0, 1)
2	[0.924, 0, 0.383, 0]	$\pi/4$	(0, 1, 0)
3	[0.707, 0.707, 0, 0]	$\pi/2$	(1, 0, 0)
4	[0.707, 0.211, 0.422, 0.527]	$\pi/2$	(0.298, 0.596, 0.745)

Table 2: Control points founded by the key knots.

Key knot (P_i)	Control point (d_i)
P_1 (0, 0, 1.047)	d_0 (0.132, −0.283, 0.307) d_1 (0.132, −0.283, 0.307)
P_2 (0, 0.785, 0)	d_2 (−0.659, 1.416, −0.251)
P_3 (1.571, 0, 0)	d_3 (2.506, −0.668, −0.304)
P_4 (0.468, 0.937, 1.171)	d_5 (0.061, 1.258, 1.466) d_5 (0.061, 1.258, 1.466)

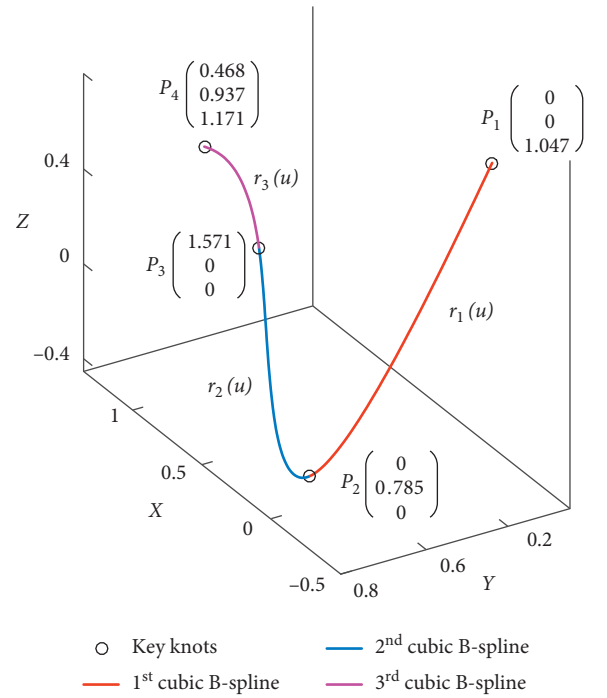


Figure 4: Interpolated curve with cubic B-spline.

When the variable u changes within the range [0,1], series of interpolated points of cubic B-spline curves can be obtained. Connecting these interpolated points, we can get three cubic B-spline curves which must pass through the key knots P_1, P_2, P_3, and P_4, as can be seen in Figure 4.

It is obvious that curve $r_1(u)$, curve $r_2(u)$, and curve $r_3(u)$ are C^3 continuous, respectively, according the above functions. It can also be verified that the whole spline composed of these three B-spline curves is C^2 continuous at the key knots P_2 and P_3. As seen in function (27), the second derivative of $r_1(u)$ with $u = 1$ is equal to the second derivative of $r_2(u)$ with $u = 0$. That is, the whole spline is C^2 continuous at the knots P_2. Similarly, the second derivative of $r_2(u)$ with $u = 1$ is equal to the second derivative of $r_3(u)$ with $u = 0$ in function (28), and the whole spline is C^2 continuous at the knots P_3. Thus, the whole B-spline $r(u)$ is C^2 continuous:

$$\left.\frac{\mathrm{d}^2[r_1(u)]}{\mathrm{d}u^2}\right|_{u=1} = \left.\frac{\mathrm{d}^2[r_2(u)]}{\mathrm{d}u^2}\right|_{u=0} = \begin{pmatrix} 3.957 \\ -3.784 \\ 1.505 \end{pmatrix}, \quad (27)$$

$$\left.\frac{\mathrm{d}^2[r_2(u)]}{\mathrm{d}u^2}\right|_{u=1} = \left.\frac{\mathrm{d}^2[r_3(u)]}{\mathrm{d}u^2}\right|_{u=0} = \begin{pmatrix} -5.610 \\ 4.011 \\ 1.822 \end{pmatrix}. \quad (28)$$

According to equation (5), the interpolated quaternion can be marked as the below equation to explain the continuity of interpolation quaternion clearly:

$$Q(u) = e^{r(u)}. \quad (29)$$

In function (29), on the left side of the equation is the interpolated quaternions. On the right side of the equation is the exponential form of interpolated B-spine. Exponent is taken for $r(u)$, and the new function obtained would have the same order continuity as $r(u)$. Because $r(u)$ in the case study described above is C^2 continuous, especially at the key knots P_2 and P_3 showed above, $\exp[r(u)]$ is surely also C^2 continuous at same knots points as checked by following equations:

$$\left.\frac{\mathrm{d}^2\left[e^{\mathbf{r}_1(u)}\right]}{\mathrm{d}u^2}\right|_{u=1} = \left.\frac{\mathrm{d}^2\left[e^{\mathbf{r}_2(u)}\right]}{\mathrm{d}u^2}\right|_{u=0} = \begin{pmatrix} 5.366 \\ -8.217 \\ 2.153 \end{pmatrix}, \quad (30)$$

$$\left.\frac{\mathrm{d}^2\left[e^{\mathbf{r}_2(u)}\right]}{\mathrm{d}u^2}\right|_{u=1} = \left.\frac{\mathrm{d}^2\left[e^{\mathbf{r}_3(u)}\right]}{\mathrm{d}u^2}\right|_{u=0} = \begin{pmatrix} -26.364 \\ 4.017 \\ 2.559 \end{pmatrix}. \quad (31)$$

The right side of equation (29), $\exp[r(u)]$, is C^2 continuous. Then, the left side of equation (29), $Q(u)$, must be C^2 continuous because the continuity on the left and right sides of the equation is the same. So, the interpolated quaternions are C^2 continuous, which is demonstrated with detail in Section 6.

(4) According to formula (11), these interpolated points of B-spline curves in R^3 Cartesian space can be mapped to the unit quaternions in S^3 space. Thus, we eventually complete the orientation interpolations of unit quaternion. Taking $u = 0$, 0.2, 0.5, 0.8, and 1 partly as examples, the relevant data of the quaternion interpolations are listed in Tables 3–5, respectively.

As can been seen in Table 3, when the variable u is equal to 0 and 1, respectively, the interpolated quaternion is [0.866, 0, 0, 0.5] and [0.924, 0, 0.383, 0] correspondingly, which are precisely the values of the quaternions corresponding to the first and second key orientations.

In Table 4, when the variable u is equal to 0 and 1, respectively, the interpolated quaternion is [0.924, 0, 0.383, 0] and [0.707, 0.707, 0, 0], which are precisely the values of the quaternions corresponding to the second and third key orientations.

Similarly, in Table 5, when the variable u is equal to 0 and 1, respectively, the interpolated quaternion is [0.707, 0.7071, 0, 0] and [0.707, 0.211, 0.422, 0.527] which are precisely the values of the quaternions corresponding to the third and fourth key orientations. The results are consistent with the expected values of interpolation theory proposed.

When we take evenly 50 times variable u within the range of [0,1], we can get 150 unit quaternions among the three segments of interpolation. In consideration of paper space, only partial interpolation data are listed in the above tables. By now, quaternion orientation interpolation ends.

6. Comparison and Discussion

In order to compare with the proposed quaternion interpolation methodology, the most commonly used quaternion interpolation methods SLERP and SQUAD described in the previous section are applied for the same case study above as well. The results of the research and analysis below reflect there are more or less deficiencies for these two methods to be used in multiorientation quaternion interpolation. The comparison and analysis mainly focus on these items: quaternion interpolation curve, rotation angle routing, angular velocity routing, angular acceleration routing, and rotating axis trajectory. The detailed comparison and analysis process are as follows.

6.1. Quaternion Interpolation Curves by Proposed Methodology, SQUAD and SLERP. Unit quaternion belongs to S^3 space and can be turned by *Hopf Mapping* to the vector in S^2 space [23], which is on the surface of unit sphere in Cartesian space. Then, the interpolated unit quaternions of end-effector orientation can be plotted graphically for observation. By this way, we can intuitively draw the interpolation trajectory diagram for the interpolated four-dimensional unit quaternions on the unit sphere in three-dimensional space. Figure 5 shows the quaternion interpolation curves for the above study case obtained by the methods of proposed methodology, SQUAD and SLERP, respectively.

In Figure 5(a), we can see the whole quaternion interpolation curve by proposed methodology is continuous and smooth. There is neither sharp point nor broken point even at the segment connecting points k_2 or k_3, which means the proposed multiorientation quaternion interpolation is reasonable. The same is shown in Figure 5(b) by the SQUAD method. However, we cannot tell whether the proposed methodology or SQUDA is C^2 continuous only from the quaternion interpolation curves. Because both of their quaternion interpolation curves seem smooth enough in appearance, further analysis and discussion are needed in below section.

As for the method SLERP, the situation becomes a little simpler. We can clearly see that sharp points occur at the

Table 3: The first B-spline curve is mapped to unit quaternions (data partly).

SN	Variable	Position of B-spline in Cartesian space			Rotation angle	Vector of rotation axis			Unit quaternion			
									Real part		Imaginary part	
	u	x	y	z	θ	a	b	c	q_0	q_1	q_2	q_3
1	0	0	0	1.047	`1.047	0	0	1	0.866	0	0	0.5
2	0.2	−0.079	0.175	0.885	0.905	−0.088	0.193	0.977	0.899	−0.038	0.085	0.427
3	0.5	−0.194	0.503	0.550	0.770	−0.251	0.653	0.715	0.927	−0.094	0.245	0.268
4	0.8	−0.174	0.746	0.208	0.794	−0.219	0.940	0.262	0.922	−0.085	0.363	0.101
5	1	0	0.785	0	0.785	0	1	0	0.924	0	0.383	0

Table 4: The second B-spline curve is mapped to unit quaternions (data partly).

SN	Variable	Position of B-spline in Cartesian space			Rotation angle	Vector of rotation axis			Unit quaternion			
									Real part		Imaginary part	
	u	x	y	z	θ	a	b	c	q_0	q_1	q_2	q_3
1	0	0	0.785	0	0.785	0	1	0	0.924	0	0.383	0
2	0.2	0.269	0.697	−0.120	0.757	0.355	0.921	−0.159	0.929	0.131	0.340	−0.059
3	0.5	0.849	0.401	−0.207	0.962	0.883	0.417	−0.216	0.887	0.409	0.193	−0.100
4	0.8	1.373	0.101	−0.145	1.384	0.992	0.073	−0.105	0.770	0.633	0.046	−0.067
5	1	1.571	0	0	1.571	1	0	0	0.707	0.707	0	0

Table 5: The third B-spline curve is mapped to unit quaternions (data partly).

SN	Variable	Position of B-spline in Cartesian space			Rotation angle	Vector of rotation axis			Unit quaternion			
									Real part		Imaginary part	
	u	x	y	z	θ	a	b	c	q_0	q_1	q_2	q_3
1	0	1.571	0	0	1.571	1	0	0	0.707	0.707	0	0
2	0.2	1.553	0.045	0.181	1.564	0.993	0.029	0.115	0.710	0.700	0.020	0.081
3	0.5	1.246	0.315	0.556	1.400	0.890	0.225	0.397	0.765	0.573	0.145	0.256
4	0.8	0.782	0.689	0.940	1.403	0.557	0.491	0.670	0.764	0.360	0.317	0.432
5	1	0.468	0.937	1.171	1.571	0.298	0.596	0.745	0.707	0.211	0.422	0.527

segment connecting points k_2 or k_3 in Figure 5(c) by SLERP. This result fully shows that the quaternion interpolation of SLERP for multiple orientations is not smooth despite being continuous and will bring violent impacts and shocks to end-effector. So, we can immediately come to a conclusion that SLERP is not suitable for multiorientation quaternion interpolation.

6.2. Rotation Angle Routings by Proposed Methodology, SQUAD and SLERP. We can get the values of rotation angle, angular velocity, and angular acceleration according to these three interpolation formulas above. Figure 6 shows rotation angle routing curves of the above study case obtained by the methods of proposed methodology, SQUAD and SLERP, respectively. The angle curve in Figure 6(c) has sharp points θ_{c1} and θ_{c2}, which implies the curve is continuous but not smooth enough and rotation angle arrangement is not reasonable. The defect led by unsmooth rotation angle planning of multiorientation interpolation of SLERP is more obvious in later analysis.

The curves in Figures 6(a) and 6(b) are very similar in appearance. Both of them look smooth and continuous, which means we cannot judge from rotation angle routing curves which one is the better choice in proposed methodology and SQUAD. Analysis below is needed.

6.3. Angular Velocity Routings by Proposed Methodology, SQUAD and SLERP. Angular velocity routing curves of these three methods for the case are shown in Figure 7. There are very obvious broken angular velocity points ω_{c1} and ω_{c2} in Figure 7(c), which shows clearly that jump changes take place in angular velocity and will bring violent impacts and shocks harmful to industrial robots. Figure 7(c) demonstrates exactly once more that SLERP is not suitable for multiorientation quaternion interpolation.

The angular velocity routing curves in Figure 7(a) and 7(b) are still similar. Because both of them are continuous and have no obvious sharp point, it is still difficult to determine which one is better in the proposed methodology and SQUAD by angular velocity.

6.4. Angular Acceleration Routings by Proposed Methodology, SQUAD and SLERP. Angular acceleration routing curves of these three methods for the case are shown in Figure 8. The angular acceleration routing curve in Figure 8(a) obtained by proposed methodology is continuous while the curve in Figure 8(b) by SQUAD is broken by the points ε_{c1} and ε_{c2}. In other words, the quaternion interpolation SQUAD for multiple orientations is not C^2 continuous and the angular acceleration is discontinuous, which will also lead to violent shocks and impacts more or less and degrade the working

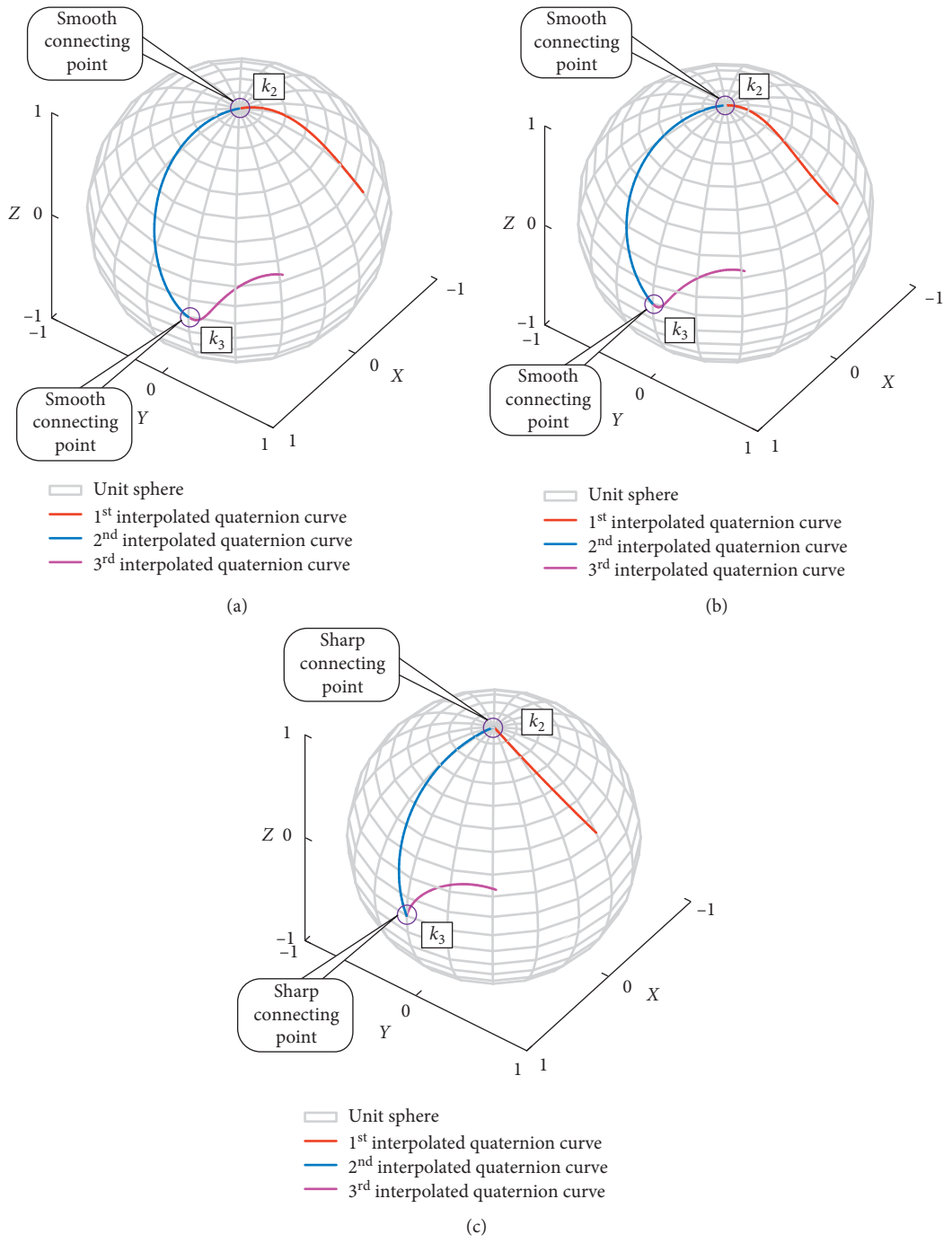


Figure 5: Quaternion interpolation curves. (a) Quaternion interpolation curves by proposed methodology. (b) Quaternion interpolation curves by SQUAD. (c) Quaternion interpolation curves by SLERP.

performance and service life of industrial robot as well. Therefore, SQUAD is not perfect quaternion interpolation for multiple orientation. The angular acceleration routing curves in Figure 8(c) obtained by SLERP are discontinuous expectedly. The previous analysis has made it very clear that SLERP is not suitable for multiorientation quaternion interpolation.

Up to now, we can tell evidentially that the proposed methodology is superior to SLERP and SQUAD in multi-orientation quaternion interpolation.

6.5. Rotation Axis Trajectories by Proposed Methodology, SQUAD and SLERP. We can also observe which method is more advantageous from rotation axis trajectories as shown in Figure 9. The rotation axis trajectory curve is from the starting orientation to the ending orientation of end-effector during the whole orientation planning.

The rotation axis trajectory by the proposed method in Figure 9(a) is smooth enough. Although the rotation axis trajectory curve by SQUAD in Figure 9(b) is continuous and smooth, we can still observe that the rotation

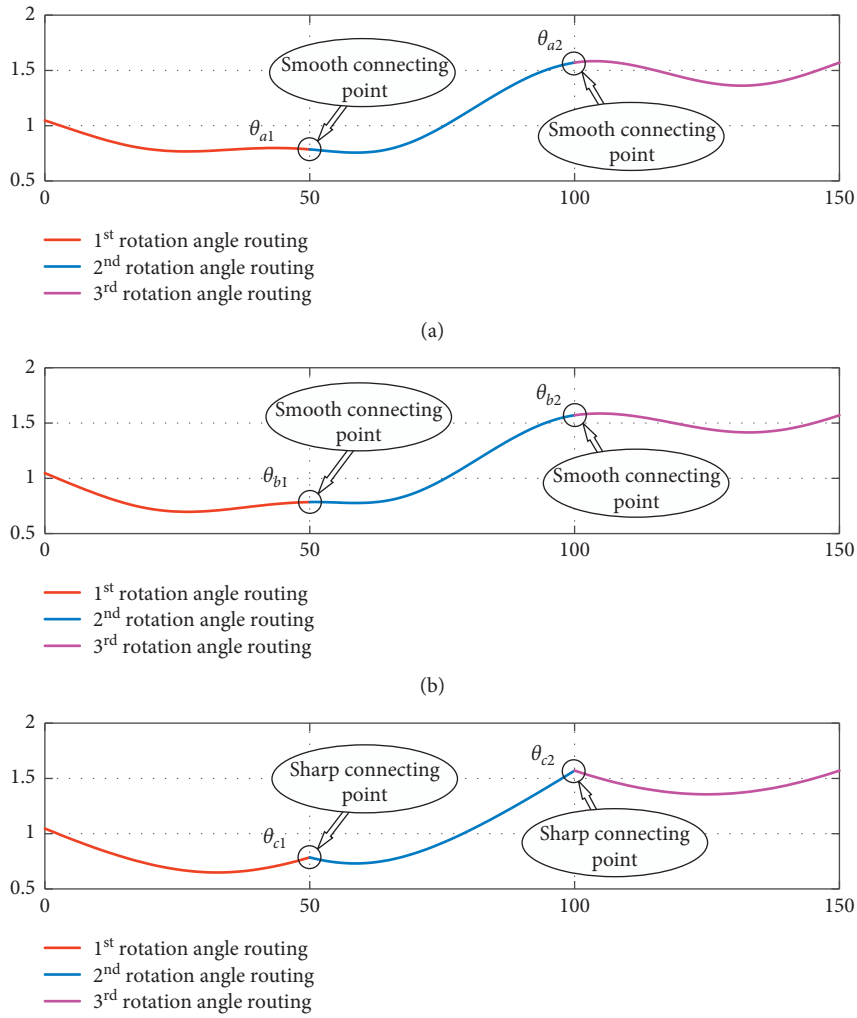


Figure 6: Angular routings. (a) Angular routing by proposed methodology. (b) Angular routing by SQUAD. (c) Angular routing by SLERP.

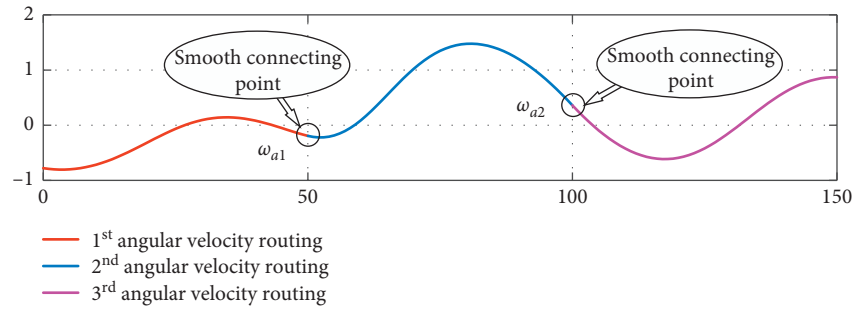


Figure 7: Continued.

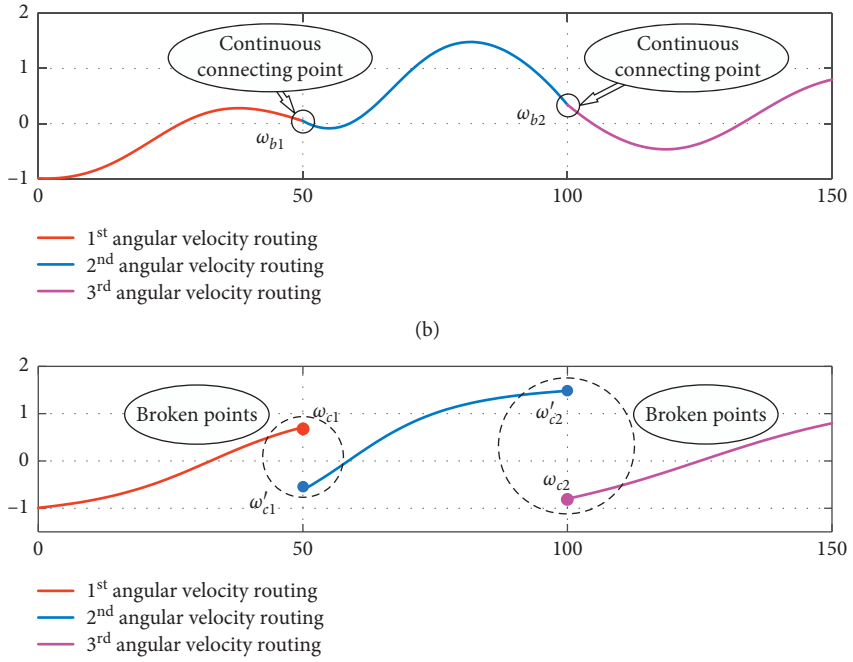


Figure 7: Angular velocity routings. (a) Angular velocity routing by proposed methodology. (b) Angular velocity routing by SQUAD. (c) Angular velocity routing by SLERP.

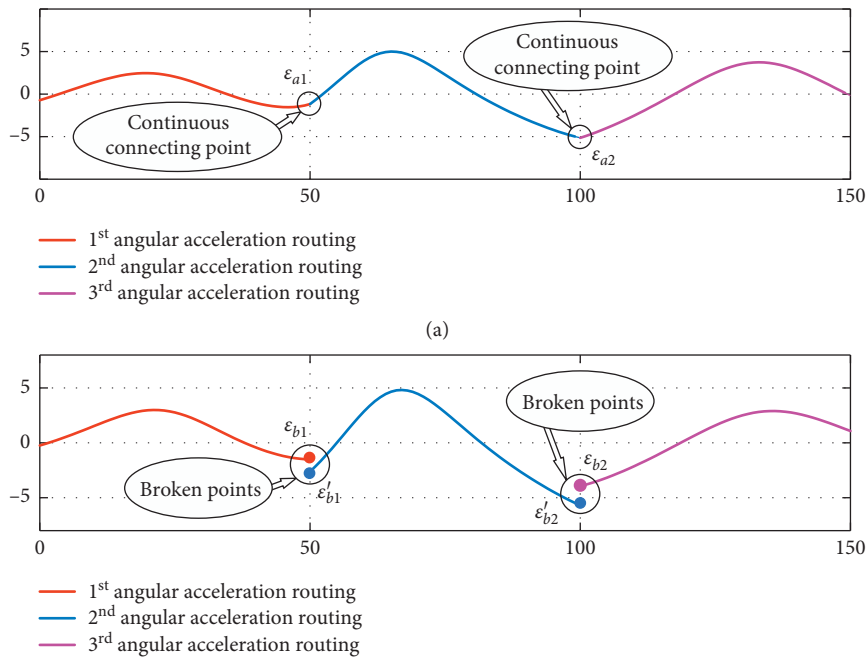
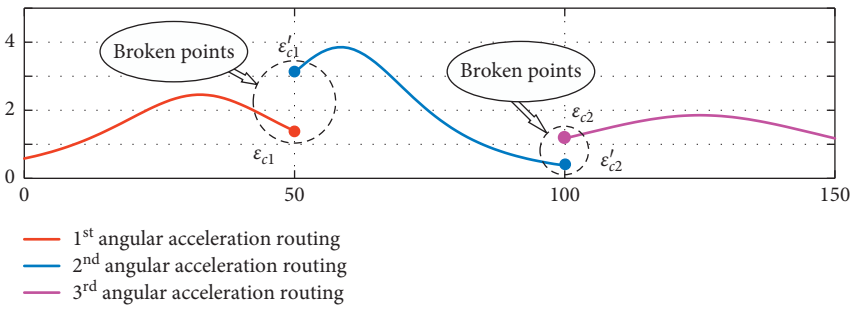
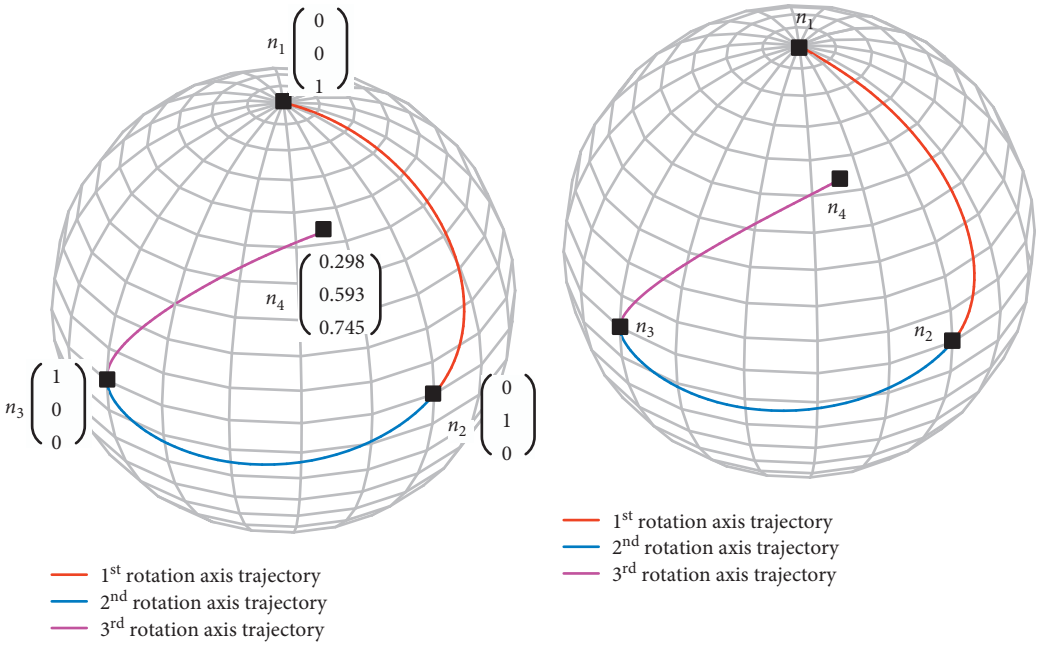


Figure 8: Continued.



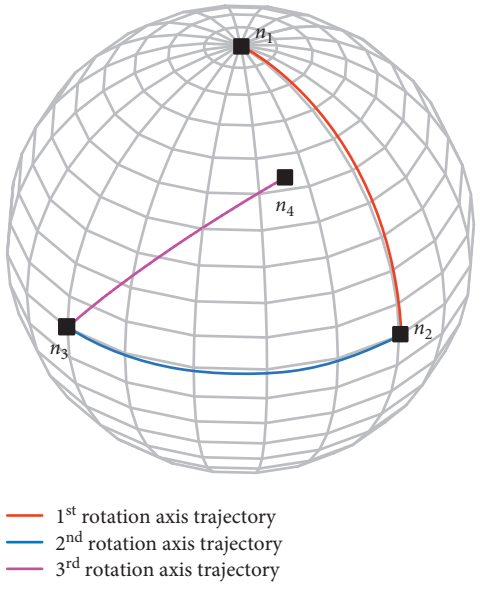
(c)

Figure 8: Angular acceleration routings. (a) Angular acceleration routing by proposed methodology. (b) Angular acceleration routing by SQUAD. (c) Angular acceleration routing by SLERP.


n_1 0 0 1
n_2 0 1 0
n_3 1 0 0
n_4 0.298 0.593 0.745
1st rotation axis trajectory
2nd rotation axis trajectory
3rd rotation axis trajectory

(a)

(b)



(c)

Figure 9: Rotation axis trajectories. (a) Rotation axis trajectory by proposed methodology. (b) Rotation axis trajectory by SQUAD. (c) Rotation axis trajectory by SLERP.

axis trajectory curve by proposed methodology in Figure 9(a) is more fluid. Moreover, the rotation axis trajectory by SLERP in Figure 9(c) has apparent sharp points n_2 and n_3 at the connecting corners and is obviously not smooth. It can be concluded as well that SLERP is not suitable for multiorientation quaternion interpolation, and proposed methodology is more advantageous than SQUAD and SLERP in multiorientation quaternion interpolation.

7. Conclusions

High-order continuous orientation planning is beneficial to working performance and service life of robot end-effector. However, the most commonly used quaternion interpolation algorithms, SLERP and SQUAD, cannot guarantee C^2 continuity of multiorientation interpolation planning. When SLERP is applied for quaternion multiorientation interpolation, it will generate intermittent angular velocity and angular acceleration at the connecting point, which will bring additional impacts and vibrations to the robot and is very harmful to the working performance of robots. When SQUAD is employed for quaternion multiorientation interpolation, it will also produce discontinuous angular acceleration at the connection point, which will also lead to additional shocks and vibrations more or less and are not conducive to the robot's work either. Therefore, SLERP and SQUAD are not suitable for quaternion multiorientation planning of robot end-effector.

The proposed quaternion interpolation method with the cubic B-spline curve based on logarithmic quaternion is able to guarantee C^2 continuity. When the proposed method is applied to the multiorientation planning of robot end-effector, the planned angular velocity and angular acceleration are both continuous and smooth avoiding additional impacts and vibrations unfavourable for robot working. By the way, in the procedure of reversely finding control points of the cubic B-spline curve, the constraint function condition is that the curvatures at the beginning and the terminal knots of the B-spline curve are set to zero. It helps the quaternion interpolation become smoother and sleeker. Moreover, the formula expression of the proposed methodology is concise and its application is simple and convenient. The study case verified the C^2 continuity of quaternion interpolation and its rationality. Therefore, it is suitable for quaternion multiorientation planning of robot end-effector. Furthermore, the proposed methodology can be extended to multiorientation quaternion interpolation with higher continuity than the second order.

Data Availability

The study case data used to support the findings of this study are available within the article and can be obtained upon request to the corresponding author as well.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant no. 51675439) and the National Science and Technology Major Projects of China (2015ZX04001003).

References

[1] H. L. Pham, B. V. Adorno, V. Perdereau et al., "Set-point control of robot end-effector pose using dual quaternion feedback," *Robotics and Computer-Integrated Manufacturing*, vol. 52, pp. 100–110, 2018.

[2] X. F. Fraisse, "Optimal pose trajectory planning for robot manipulators," *Mechanism and Machine Theory*, vol. 37, no. 10, pp. 1063–1086, 2002.

[3] P. T. Zacharia, E. K. Xidias, and N. A. Aspragathos, "Task scheduling and motion planning for an industrial manipulator," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 6, pp. 449–462, 2013.

[4] A. M. Zanchettin and P. Rocco, "Motion planning for robotic manipulators using robust constrained control," *Control Engineering Practice*, vol. 59, no. 59, pp. 127–136, 2017.

[5] K. Kaltsoukalas, S. Makris, and G. Chryssolouris, "On generating the motion of industrial robot manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 32, no. 32, pp. 65–71, 2015.

[6] P. B. Mulik, "Optimal trajectory planning of industrial robot with evolutionary algorithm," in *Proceedings of the 2015 International Conference On Computation Of Power, Energy, Information And Communication (ICCPEIC)*, IEEE, Chennai India, pp. 0256–0263, April 2015.

[7] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[8] J. Coburn and J. J. Crisco, "Interpolating three-dimensional kinematic data using quaternion splines and hermite curves," *Journal of Biomechanical Engineering*, vol. 127, no. 2, pp. 311–317, 2005.

[9] K. Shoemake, "Animating rotation with quaternion curves," *ACM SIGGRAPH Computer Graphics*, vol. 19, no. 3, pp. 245–254, 1985.

[10] E. B. Dam, M. Koch, and M. Lillholm, "Quaternions, interpolation and animation," vol. 2, Datalogisk Institut, Københavns Universitet, Copenhagen, Denmark, 1998.

[11] M. J. Kim, M. S. Kim, and S. Y. Shin, "A general construction scheme for unit quaternion curves with simple high order derivatives," in *Proceedings of the September 1995 SIGGRAPH*, vol. 95, ACM, New York, NY, USA, pp. 369–376, September 1995.

[12] Y. Xing, R.-z. Xu, J.-q. Tan et al., "A class of generalized B-spline quaternion curves," *Applied Mathematics and Computation*, vol. 271, pp. 288–300, 2015.

[13] X. Fan and T. Wang, "C2-continuous orientation trajectory planning for robot based on spline quaternion curve," *Assembly Automation*, vol. 38, no. 3, pp. 282–290, 2018.

[14] W. Ge, Z. Huang, and G. Wang, "Interpolating solid orientations with a C2-continuous B-spline quaternion curve," in *Proceedings of the International Conference On Technologies For E-Learning And Digital Entertainment*, Springer, Berlin, Germany, pp. 606–615, July 2007.

[15] P. Li and R. Guo, "Research on quaternion-quintic spherical Bezier spline interpolation algorithm for 5-axis machining," in *Proceedings of the 2009 4th International Conference on*

Computer Science & Education, IEEE, Nanning, China, pp. 612–617, July 2009.

[16] B. B. Salmerón-Quiroz, J. G. Castellanos, S. R. Paredes et al., "Global estimation of robot's attitude via quaternion and data fusion," in *Proceedings of the 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, IEEE, Singapore, Singapore, pp. 524–529, July 2012.

[17] K. T. Miura, "Unit quaternion integral curve: a new type of fair free-form curves," *Computer Aided Geometric Design*, vol. 17, no. 1, pp. 39–58, 2000.

[18] J. Tan, Y. Xing, W. Fan et al., "Smooth orientation interpolation using parametric quintic-polynomial-based quaternion spline curve," *Journal of Computational and Applied Mathematics*, vol. 329, pp. 256–267, 2018.

[19] K. Hong, "Singularities of the euler wrist," *Mechanism and Machine Theory*, vol. 35, no. 4, pp. 505–515, 2000.

[20] C. De Boor, "On calculating with B-splines," *Journal of Approximation Theory*, vol. 6, no. 1, pp. 50–62, 1972.

[21] Z. Chang and Y. Wan, *Computer Aided Geometric Modeling Technology,"* pp. 63–68, Science Press, Beijing, China, 3rd edition, 2013.

[22] Z. Shi, *"Computer Aided Geometric Design and Non-uniform Rational B-Spline (CAGD & NURBS),"* pp. 221–225, Beijing University of Aeronautics and Astronautics Press, Beijing, China, 2013.

[23] L. H. Ryder, "Dirac monopoles and the Hopf map S3 to S2," *Journal of Physics A: Mathematical and General*, vol. 13, no. 2, pp. 437–447, 1980.